



Australian Government

PCT/AU2004/001744

Patent Office
Canberra

I, JANENE PEISKER, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. 2003906918 for a patent by CANON INFORMATION SYSTEMS RESEARCH AUSTRALIA PTY LTD as filed on 12 December 2003.



WITNESS my hand this
Twenty-fourth day of December 2004

A handwritten signature in black ink, appearing to read "J. Peisker".

JANENE PEISKER
TEAM LEADER EXAMINATION
SUPPORT AND SALES

BEST AVAILABLE COPY

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION FOR THE INVENTION ENTITLED:

Efficient Whole Page Printing

Name and Address of Applicant:

Canon Information Systems Research Australia Pty Ltd, an Australian Company,
ACN 003 943 780, of 1 Thomas Holt Drive, North Ryde, New South Wales,
2113, Australia

Names of Inventors:

Richard Cudd and Jeremy David Michael Thorp and John Stewart Reeves

This invention is best described in the following statement:

EFFICIENT WHOLE PAGE PRINTING

Field of the Invention

The present invention relates to printing documents, such as HTML web page documents and, in particular, to the printing and scaling of documents to fit the nearest
5 whole page.

Background

Many computer based systems exist which are designed for the creation and viewing of documents. Many of these systems also provide mechanisms for printing documents. Word processing applications typically display a view of a document as it would appear on
10 a printed page whilst the user is still in the editing process. Most will require the user to select a paper size in which to create the document. This allows the user to obtain a feel for how a document would look when printed whilst the document is in the process of being created. At the very least, many word processors will provide an indication of where one printed page will finish and another will begin. Since these applications create documents
15 to fit a particular paper size, there is never any need for scaling or resizing of pages before printing.

Other applications such as spreadsheet tools enable the creation of documents which are not necessarily specifically designed to be printed. A user can see a preview of how the spreadsheet would look on a printout but they are not required to create the document to fit
20 a paper size. In this case the user may create a document that would not entirely fit on a particular paper size or may unnecessarily overlap by a very small amount onto a final page and thus waste paper. This problem has been addressed to some degree by Microsoft® Excel™, a spreadsheet application manufactured by Microsoft Corporation. Excel™

provides the ability to fit a spreadsheet to X pages wide by Y pages tall when printing. This will fit the spreadsheet correctly to the number of pages preferred by the user.

Web pages are predominately designed to be viewed within a web browser window on a computer display and are thus not optimized for printing. When printing web pages, it is common that content may be clipped on the right hand side of the page because the content is too wide for the paper size. Some applications, such as Netscape Navigator®, manufactured by Netscape Communications Corporation, and Canon® Easy-WebPrint™ (see www.canoneasywebprint.com) apply automatic scaling to the content to prevent this clipping and fit the all the content on the page. However, this does not solve the problem of the web page content overlapping onto a final page by only a very small amount. An example of this is shown in Fig. 4 where a document 400 has been printed with content filling a first page 402, but flow over onto only a first line of a second page 404. Products such as Canon® Easy-WebPrint™ 2.0 and Netscape Navigator® 7.0 provide the ability to manually scale the content of web pages when printing. This allows the user to repeatedly reduce the size of the content until the content fits onto the nearest whole page. This approach requires manual intervention.

Summary of the Invention

It is an object of the present invention to substantially overcome or at least ameliorate one or more problems associated with existing arrangements. The present disclosure affords methods of detecting a small overlap and scaling the web page to remove the overlap.

In accordance with one aspect of the present invention, there is disclosed a method of printing a document sourced from a computer network and spanning a plurality of printable pages, the method comprising the steps of:

determining whether an amount of content on a last page of the printable pages is less than a predetermined amount; and

when the amount of content is less than the predetermined amount, scaling the content down to fit the content to the nearest whole page and printing the content.

5 According to another aspect of the present invention, there is disclosed a printing application program arranged for printing a document sourced from a computer network and spanning a plurality of printable pages, the program comprising:

code arranged to determine whether an amount of content on a last page of the printable pages is less than a predetermined amount; and

10 code, operable when the amount of content is less than the predetermined amount, for scaling down the content to fit the content to the nearest whole page and printing the content. The computer network typically comprises the World Wide Web and the printing application program is configured to interact with a web browser application program having an associated graphical user interface, the printing application program further
15 comprising code arranged to display via the graphical user interface both a print preview of the document spanning the plurality of pages and a print preview of the document scaled to the nearest whole page.

Numerous other aspects of the present invention are also disclosed.

Brief Description of the Drawings

20 At least one embodiment of the present invention will now be described with reference to the drawings and appendices, in which:

Fig. 1 shows a modified toolbar used in the Internet Explorer™ web browser application incorporating the custom print option of the present disclosure;

Fig. 2 is a schematic block diagram of a general purpose computer upon which arrangements described can be practiced;

Fig. 3 depicts a custom printing approach for Internet Explorer™;

Fig. 4 shows an example of a print which overflowed onto a second page;

5 Fig. 5 shows the print template element behaviours;

Fig. 6 shows a prior art process of creating a print with a print template;

Fig. 7 shows a flow chart of a modified printing process;

Fig. 8 shows the threshold LAYOUTRECT;

10 Fig. 9 shows a flow chart of the modified printing process including the dialog box prompt;

Fig. 10 shows a print preview with the normal print at the top and the scaled print at the bottom;

Fig. 11 shows a print preview with the print type selected by highlighting;

Fig. 12 shows a print preview with the print type selected by right clicking;

15 Fig. 13 shows an interactive approach for varying the threshold LAYOUTRECT; and

Fig. 14 shows a dialog approach for varying the LAYOUTRECT threshold.

Detailed Description including Best Mode

20 The present disclosure proposes a printing application, termed herein as "custom print", which is preferably implemented as an add-on toolband to a generic web browser application, such as Internet Explorer™ 5.5 or 6.0, manufactured by Microsoft Corporation. An example of this is seen in Fig. 1 where part of the Internet Explorer™ 6.0 graphical user interface (GUI) 100 is shown. The GUI 100 includes a known first toolband 102 corresponding to that which permits access to the Google™ search engine and associated

facilities. A further toolband 104, developed for customised printing as part of the present disclosure, is provided to enable customised printing of web pages. It is observed that the custom print toolband 104 is distinct from a generic print icon 106 contained in the toolband 108 of the GUI 100. In this connection, the selection of the print icon 106 will
5 result in traditional printing of the web page displayed by the GUI 100 according to Internet Explorer™ 6.0, whereas selection of a print icon 110 or a print preview icon 112 found within the custom print toolband 104 will enable printing in the fashion now to be described with reference to the remaining drawings.

The method of customised printing is preferably practiced using a general-purpose
10 computer system 200, such as that shown in Fig. 2 wherein the processes of Figs. 3 to 12 may be implemented as software, such as an application program executing within the computer system 200. In particular, the steps of method of customised printing are effected by instructions in the software that are carried out by the computer. The instructions may be formed as one or more software code modules, each for performing one or more
15 particular tasks. The software may also be divided into three separate parts, in which a first part that performs the browsing methods, a second part that performs the customised printing methods and a third part that manages a user interface between the first part and second parts and the user. The third part incorporates the GUI 100 including the toolband 102 and the icons 110 and 112. The software may be stored in a computer
20 readable medium, including the storage devices described below, for example. The software is loaded into the computer from the computer readable medium, and then executed by the computer. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer preferably effects an advantageous apparatus for customised

printing of individual elements of structured documents such as web pages. The application program for customised printing may operate in concert with a web browser application also operating within the computer system 200.

The computer system 200 is formed by a computer module 201, input devices such as a keyboard 202 and mouse 203, output devices including a printer 215, a display device 214 and loudspeakers 217. A Modulator-Demodulator (Modem) transceiver device 216 is used by the computer module 201 for communicating to and from a communications network 220, and is connectable via a telephone line 221, for example, or other functional medium. The modem 216 can be used to obtain access to a server 522, the Internet, World Wide Web and other network systems, such as a Local Area Network (LAN) or a Wide Area Network (WAN), and may be incorporated into the computer module 201 in some implementations.

The computer module 201 typically includes at least one processor unit 205, and a memory unit 206, for example formed from semiconductor random access memory (RAM) and read only memory (ROM). The module 201 also includes an number of input/output (I/O) interfaces including an audio-video interface 207 that couples to the video display 214 and loudspeakers 217, an I/O interface 213 for the keyboard 202 and mouse 203 and optionally a joystick (not illustrated), and an interface 208 for the modem 216 and printer 215. In some implementations, the modem 216 may be incorporated within the computer module 201, for example within the interface 208. A storage device 209 is provided and typically includes a hard disk drive 210 and a floppy disk drive 211. A magnetic tape drive (not illustrated) may also be used. A CD-ROM drive 212 is typically provided as a non-volatile source of data. The components 205 to 213 of the computer module 201, typically communicate via an interconnected bus 204 and in a manner which

results in a conventional mode of operation of the computer system 200 known to those in the relevant art. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations or alike computer systems evolved therefrom.

5 Typically, the browser application program, by which a user of the computer 200 access the Web, is resident on the hard disk drive 210 and read and controlled in its execution by the processor 205. Intermediate storage of the program and any data fetched from the network 220 may be accomplished using the semiconductor memory 206, possibly in concert with the hard disk drive 210. In some instances, the application program may be
10 supplied to the user encoded on a CD-ROM or floppy disk and read via the corresponding drive 212 or 211, or alternatively may be read by the user from the network 220 via the modem device 216. Still further, the software can also be loaded into the computer system 200 from other computer readable media. The term "computer readable medium" as used herein refers to any storage or transmission medium that participates in providing
15 instructions and/or data to the computer system 200 for execution and/or processing. Examples of storage media include floppy disks, magnetic tape, CD-ROM, a hard disk drive, a ROM or integrated circuit, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 201. Examples of transmission media include radio or infra-red
20 transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like. Actuation of the print icons 106 or 110 causes the respective print functions to effect printing using the printer 215, for example.

The modem 216 enables the user of the computer 200 to access a web page via the network 220. The web page may be resident on the server computer 222 and accessible via a Web address defined by a Uniform Resource Locator (URL) to thereby reproduce content to the user, typically via the display 214. Typically the browser application activates the
5 GUI 100 upon the display 214 by which the content and other information associated with the web page is presented.

From Internet ExplorerTM version 5.5 onwards, it has been possible to customize how that web browser application previews and prints documents. The mechanism for printing and previewing is controlled by print templates, which are HTML files that
10 developers can create to control the layout and look of a print job. The HTML files contain JScript code (a Microsoft version of JavaScriptTM, developed by Sun Microsystems and Netscape Inc.) that is used for manipulating content and accessing objects within the print template. While print templates are HTML files, they can only be applied by making calls from within code written in a programming language such as C++. This code is loaded as a
15 dynamically linked library (.dll) file and embedded in the web browser application, thereby enabling the web browser application to issue IDM_PRINT or IDM_PRINTPREVIEW commands. These are the commands that are used for normal printing, such as via the icon 106, and print previewing by Internet ExplorerTM and provide the path to the new print template. Browsers traditionally interpret the source HTML file by parsing the HTML code
20 to create the corresponding DOM which, of itself, models the hierarchical structure of the web page. The DOM can be used to both manipulate and process the currently loaded document.

Fig. 3 demonstrates how custom print templates are called from C++ code. A method 300 of processing printing commands is shown which operates as part of the

GUI 100. In a first step 302, the browser application detects and determines a user selection of print or print preview from the GUI 100. Where default Internet Explorer™ printing is selected, for example via the icon 106, step 304 allows Internet Explorer™ to call its own default print template so as to print in a traditional fashion. Where step 302 detects
5 selection of custom printing via the icons 110 or 112, step 306 follows where the C++ .dll code embedded in Internet Explorer™ issues a corresponding IDM_PRINT or IDM_PRINTPREVIEW command. Step 308 then follows where the embedded code triggers the custom print template for HTML pages.

By creating print templates and toolbands, it is possible to control:

- 10 (i) the layout of pages when printed/previewed, and the content that is printed/previewed on them;
- (ii) how print jobs are handled - for instance, which pages are printed, and in what order; and
- (iii) the look of the print preview window and controls available on the
15 print preview user interface.

A print template in Internet Explorer™ is written using standard HTML, JScript and four element behaviours specific to print templates, those being:

- DEVICERECT
- LAYOUTRECT
- 20 • TEMPLATEPRINTER
- HEADERFOOTER

These four new element behaviors are used within print templates in the following fashion:

DEVICERECT - A DEVICERECT element represents a page to be printed or viewed in print preview. The number of physical pages printed will correspond to the number of DEVICERECT elements within the print template. The DEVICERECT will have width and height properties that correspond to the width and height of the physical page. These are obtained by querying the TEMPLATEPRINTER element.

LAYOUTRECT - LAYOUTRECT elements define the area or areas (and their styles) on a page where a document's (web page) content is displayed when printed or during print preview. In a print template, LAYOUTRECT elements are contained by the DEVICERECT elements described previously. A DEVICERECT can contain more than one LAYOUTRECT. As well as width and height properties the LAYOUTRECT also has a zoom property that can be used to scale its content. This invention uses this attribute for scaling content to fit a page. To add content to a LAYOUTRECT, the contentSrc attribute can be passed a URL string. The page pointed at by the URL will then load or "flow" into the LAYOUTRECT.

TEMPLATEPRINTER - The TEMPLATEPRINTER element provides a number of methods that give a print template control over the start and end of print jobs, control over the printing of each individual page in a print job, and control over the display of printing dialog boxes, such as the standard Print and Page Setup dialogs found in versions of Internet Explorer™ and in various Windows™ operating systems, manufactured by Microsoft Corporation. The properties of TEMPLATEPRINTER enable a print template to set or retrieve the page setup settings and current print job settings. For instance, a print template might set or retrieve the page width and page height for page setup, or the start and finish pages to determine the page range to print.

HEADERFOOTER - The **HEADERFOOTER** behaviour is a conversion tool used by the print template to generate HTML from the header and footer formatting strings defined by the Page Setup dialog box. The element contains properties **textHead** and **textFoot** that can be used to insert text headers and footers on the page to be printed. This
5 often includes information such as the page URL and the page number.

The relationship between some of these elements is illustrated in Fig. 5 which shows a print preview GUI 500 and various rectangles depicting the elements discussed above.

The basic process 600 of creating a traditional web page print according to Internet Explorer™ is shown in Fig. 6. At a start step 602, a **DEVICERECT** is created for the first
10 page of the print job, and a **LAYOUTRECT** is placed upon the **DEVICERECT** to house the web page content. At step 604, the web page content is then loaded or “flowed” into the **LAYOUTRECT**. Step 606 tests if the content of the page overflows the bounding area established by this **LAYOUTRECT**. Such an overflow is generally indicated by a software event occurring within the Application Program Interface (API) associated with the Internet
15 Explorer™ print function. Step 606 as such tests for the occurrence of this software event. If no overflow occurs, then the content is able to be reproduced on the current (the first) page, and the process 600 ends at step 618. If the content overflows the **LAYOUTRECT**, as detected by the software event, a new page, being a **DEVICERECT**, is created at step 608. A new **LAYOUTRECT** is then placed upon that **DEVICERECT** at step 612 to house the
20 excess content. Control then returns to step 604 and the process continues until there is no more content left to add. In this fashion, the print job, “builds” the appropriate number of pages in order to satisfy the amount of content.

In order to provide a check to see whether the content has "only just" overlapped onto the last page of the print job, as illustrated in Fig. 4, a modified process 700 as shown in Fig. 7 may be used.

In the process 700, steps 702, 704, 706 and 708 each correspond respectively to steps 602, 604, 606 and 608 just described. However, each time a new DEVICERECT is added to the print job in step 708, and subsequently a new LAYOUTRECT is placed upon it in step 710, this LAYOUTRECT is set to be the height of a user defined or predetermined threshold. An example of this is shown in Fig. 8 where a print job 800 has two pages 802 and 804. A "threshold" LAYOUTRECT 806 is created which is limited to only a portion of the normal page height. Then, according to step 712, the content that overflows from the previous page 802 flows into this small LAYOUTRECT 806, as illustrated.

A test is performed in step 714 to determine if the content overflows the small threshold LAYOUTRECT. If so, control passes to step 722 which operates to remove the threshold LAYOUTRECT, thereby facilitating insertion at step 704 of a full-size LAYOUTRECT. As an alternative, step 722 may operate to stretch (ie. re-size) the threshold LAYOUTRECT to a new size, such as that corresponding to a full page size. If the content does not overflow the small LAYOUTRECT 806, indicating that the content is less than the size of the small LAYOUTRECT 806, then the content is considered to have overlapped onto the final page by only a small amount. If this is the case, then all the content contained in the print job is scaled down to the nearest whole page according to step 716. The scale factor used is calculated as the total height of all the whole pages divided by the total height of all the whole pages plus the threshold height.

In the example of Fig. 8, if the threshold height is 20% of the page height, then the scale factor will be $(1/(1+0.2)) = 0.83$. Step 716 alters (reduces) the size of the content by

the scale factor to fit the content onto the page 802 and the process ends at step 716. Step 716 operates to expunge the additional, now empty page 804 from the print job. The process ends at step 718.

Fig. 9 shows an alternate process 900 in which steps 902-918 and 922 each
5 correspond respectively to those of steps 702-718 and 722 described above. In this implementation, step 920, which is inserted between steps 914 and 916, operates to present the user with a dialog box, depicted at 924 in the instance where the content is determined to have overlapped onto the final page by only a small amount. The dialog box operates to prompt the user as to whether they wish to scale the content to the nearest whole page, or
10 just to continue with a normal print. If scaling is selected, step 916 follows as previously described. Alternatively the process 900 ends at step 918 and a traditional number of pages are printed.

The threshold LATOUTRECT may be altered in a number of ways. Because of the operation of the Internet ExplorerTM program, such alteration is best performed prior to the
15 commencement of the methods 700 or 900, or within the corresponding start steps 702 and 902. One reason for altering the threshold is to avoid "overscaling", which may result in content being scaled down to a size which is too small in a user's opinion, and thus negates any advantage obtained by printing the document on one less page.

One approach to such altering is to present the user of the computer 200 with a
20 dialog box 1400 as shown in Fig. 14. The dialog box 1400 is a subsidiary graphical user interface to that associated with the web browser and indicates the size 1402 of the threshold as a percentage of the page size and also provides a representation 1404 of a page illustrating the threshold 1406 corresponding to the values 1402. With this, the user may re-enter the threshold by typing over the size indication 1402. Alternative, the user may

select the threshold line 1406 from the representation 1404 using the mouse 203 in a traditional fashion and reposition the threshold 1406 by way of a drag operation, as indicated by the arrow 1408. The drag operation is constrained to be either up or down, and not sideways. During such a drag operation, the numerically represented threshold 1402
5 may automatically be updated according to the repositioning of the threshold 1406. When satisfied with a desired value of the threshold, the user may select "SET" 1410 to establish the threshold, and then permit the methods 700, 900 to proceed.

A further user interaction may be desirable where the size of the threshold LAYOUTRECT is larger than the amount of content that it is required to contain. In this
10 instance, the user may choose to reduce the size of the threshold LAYOUTRECT to a size that neatly accommodates the overflow content. Scaling may then be performed with a larger scale factor than would have been available with a fixed threshold LAYOUTRECT. The initial threshold LAYOUTRECT in such an implementation may be indicated in a print preview type display 1300 as shown in Fig. 13, by a selectable bounding box 1302
15 superimposed upon the print preview display. The bounding box 1302, when selected, by a mouse click for example, allows the user a (further) attempt to adjust the presentation of the content. The user may move the lower extremity 1304 of the bounding box 1302 in an upward direction (indicated by the arrow 1308) to a position 1306 to reduce the size of the bounding box 1302. This has the same effect of altering, in a constrained fashion (ie. by
20 reduction only), the percentage size of the threshold. Once the threshold LAYOUTRECT is reset, the methods 700 or 900 may then be implemented to achieve the desired reduction onto a smaller number of pages, but at a larger scale factor than otherwise would have been use had a default threshold size been relied upon.

Fig. 10 shows an example of a print preview GUI 1000 that may be presented to the user before actual printing occurs. The print preview 1000 includes a traditional print preview 1002 of a document spanning three printable pages 1004, 1006 and 1008 and which occupies one half of the GUI 1000. The other half of the GUI 1000 is occupied by a
5 print preview 1010 of the same document scaled according to the processes of Figs. 7 or 9 and which includes two completely filled printable pages 1012 and 1014. The print job in this case has been scaled to the nearest whole page.

With reference to Fig. 11, the user can then select in the print preview 1000 whether they would like to print the normal or scaled versions of the print job. In one
10 implementation, this selection can be made by positioning a pointer cursor 1106 associated with the mouse 203 within the desired print preview (1002 or 1010) and clicking the mouse 203. The selected type of print can then be highlighted, as illustrated in Fig. 11 for the scaled preview 1010, and will be printed upon user selection of the print button icon 1108.

15 In a further alternative, illustrated in Fig. 12, the user may perform the selection by right clicking the mouse 203 whilst the cursor 1106 is positioned on the half of the preview screen containing the type of preview desired to be printed. A popup menu 1200 will appear allowing the user to print as well as providing a number of other operations such as changing the zoom or displaying other options. These operation may each be instigated
20 from the popup menu 1200 via the GUI 1000 in a traditional fashion.

Industrial Applicability

The arrangements described are applicable to the computer and data processing industries and particularly those involving the printing of data sourced from computer

networks. The disclosure is directly applicable to web browsing applications and the printing of data sourced using such applications.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive.

For example, the implementations discussed above describe scaling the entire content to fill a whole number of pages. This is done to ensure uniformity of presentation essentially consistent with the original presentation for a traditional print. However, all pages need not be scaled. In a multi-page document, only the last "page" of content need be scaled to ensure an appropriate fit and avoid minor overflow. This however will result in a greater amount of scaling (smaller scale ratio) which may be quite evident in comparison with those page that were not subject to scaling.

(Australia Only) In the context of this specification, the word "comprising" means "including principally but not necessarily solely" or "having" or "including", and not "consisting only of". Variations of the word "comprising", such as "comprise" and "comprises" have correspondingly varied meanings.

The claims defining the invention are as follows:

1. A method of printing a document sourced from a computer network and spanning a plurality of printable pages, said method comprising the steps of:
 - 5 determining whether an amount of content on a last page of said printable pages is less than a predetermined amount; and
when the amount of content is less than the predetermined amount, scaling the content down to fit the content to the nearest whole page and printing the content.
- 10 2. A method according to claim 1 wherein the nearest whole page is the plurality less one, and if the amount of content is greater than said predetermined amount, then printing said plurality of pages.
3. A method according to claim 2, wherein if the content on the last page is less than
15 the predetermined amount, said method comprises the further step, before said scaling, of:
 - determining a user preference for one of scaling the content to fit the nearest whole page or printing said plurality of pages.
4. A method according to claim 3 wherein said determining comprises presenting a
20 dialog box to the user within a graphical user interface whereby the user can select one of scaling the content to fit the nearest whole page or printing said plurality of pages
5. A method according to claim 1, wherein said method is performed in conjunction with a graphical user interface arranged to display both a print preview of said document

spanning said plurality of pages and a print preview of said document scaled to said nearest whole page.

6. A method according to claim 5 further comprising detecting a user selection of one
5 of said print previews and printing the selected print preview.

7. A method according to claim 1 wherein said predetermined amount is user adjustable.

10 8. A method according to claim 7 further comprising presenting a graphical user interface including a value of said predetermined amount and detecting a user's change to said amount made via said graphical user interface.

8a. A method according to claim 8 wherein said graphical user interface comprise at
15 least one of an numerical representation of said value and a user manipulable graphical representation of said value.

9. A method according to claim 7 further comprising representing said predetermined amount within a bounding area in a graphical user interface and detecting manipulation of
20 said bounding area by the user, said manipulation thereby adjusting at least adjusting a scale factor for said scaling.

10. A printing application program arranged for printing a document sourced from a computer network and spanning a plurality of printable pages, said program comprising:

code arranged to determine whether an amount of content on a last page of said printable pages is less than a predetermined amount; and

code, operable when the amount of content is less than the predetermined amount, for scaling down the content to fit the content to the nearest whole page and printing the
5 content.

11. A printing application program according to claim 10 wherein the nearest whole page is the plurality less one, said program further comprising code, operable when the amount of content is greater than said predetermined amount, for printing said plurality of
10 pages.

12. A printing application program according to claim 11 wherein said computer network comprises the World Wide Web and said printing application program is configured to interact with a web browser application program having an associated
15 graphical user interface, said printing application program further comprising code arranged to display via said graphical user interface both a print preview of said document spanning said plurality of pages and a print preview of said document scaled to said nearest whole page.

20 13. A printing application program according to claim 12 further comprising code for detecting a user selection of one of said print previews and printing the selected print preview.

14. A printing application program according to claim 12 further comprising code for presenting a subsidiary graphical user interface box within said graphical user interface of said web browser application whereby a user thereof can select one of scaling the content to fit the nearest whole page or printing said plurality of pages.

5

15. A printing application program according to claim 14 wherein said subsidiary graphical user interface comprises a dialog box including a user manipulable graphical representation of predetermined amount.

10 16. A computer readable medium having recorded thereon a printing application program according to any one of claims 10 to 15.

17. A method of printing a document sourced from a computer network substantially as described herein with reference to any one of Figs. 7 to 14.

15

18. A computer apparatus incorporating the invention of any one of the preceding claims.

DATED this TWELTH Day of DECEMBER 2003

20

CANON INFORMATION SYSTEMS RESEARCH AUSTRALIA PTY LTD

Patent Attorneys for the Applicant

SPRUSON&FERGUSON

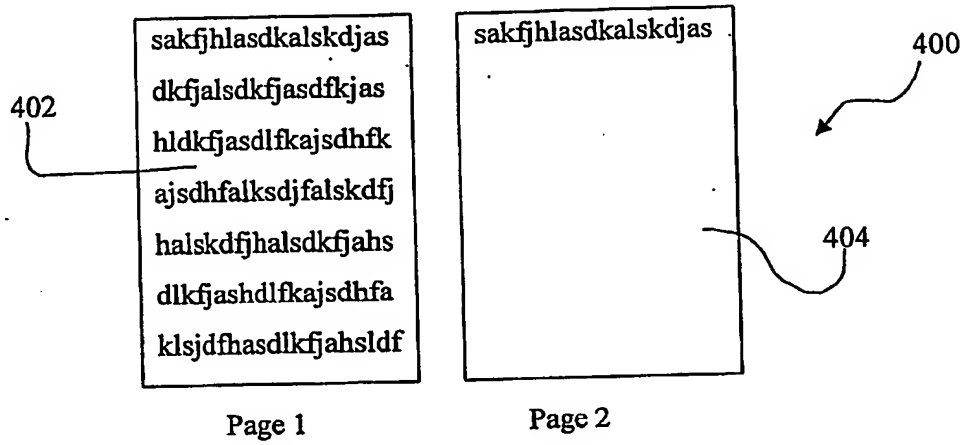
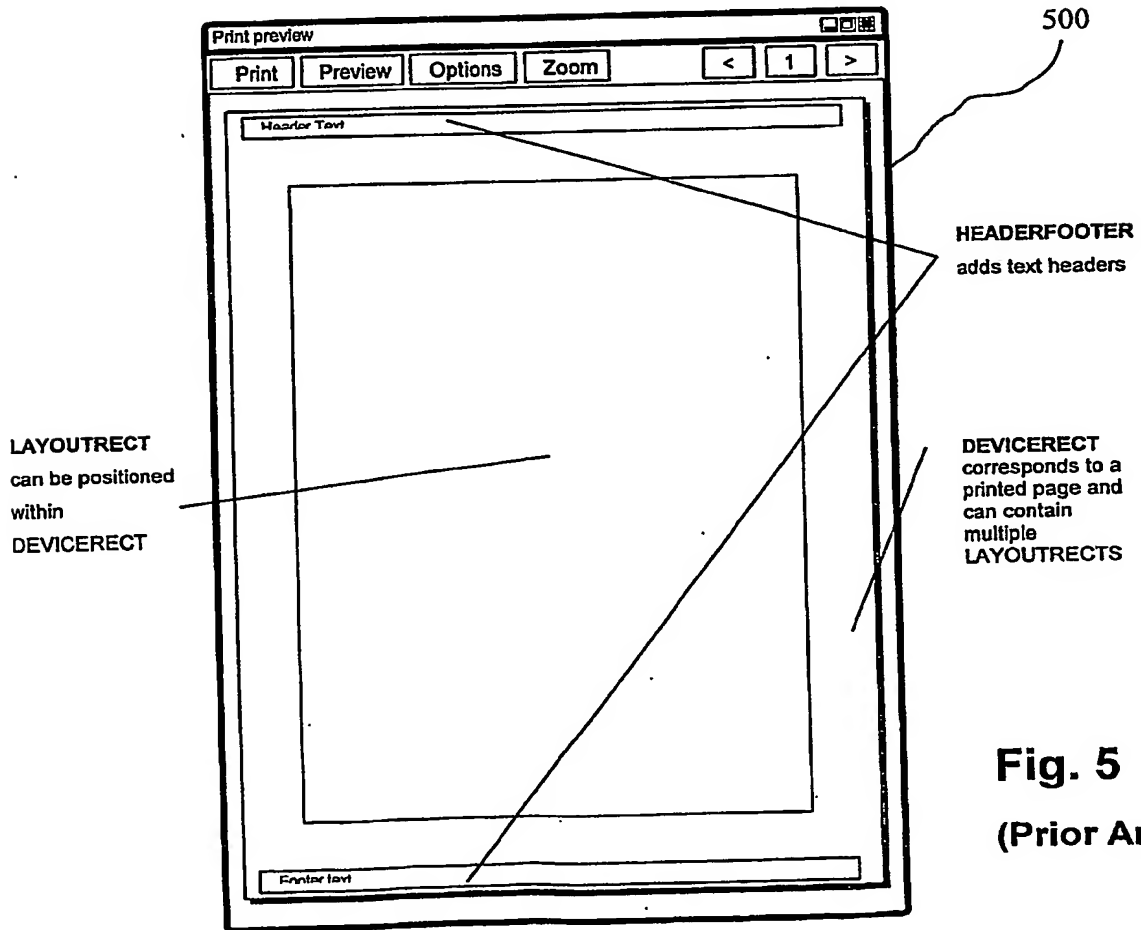
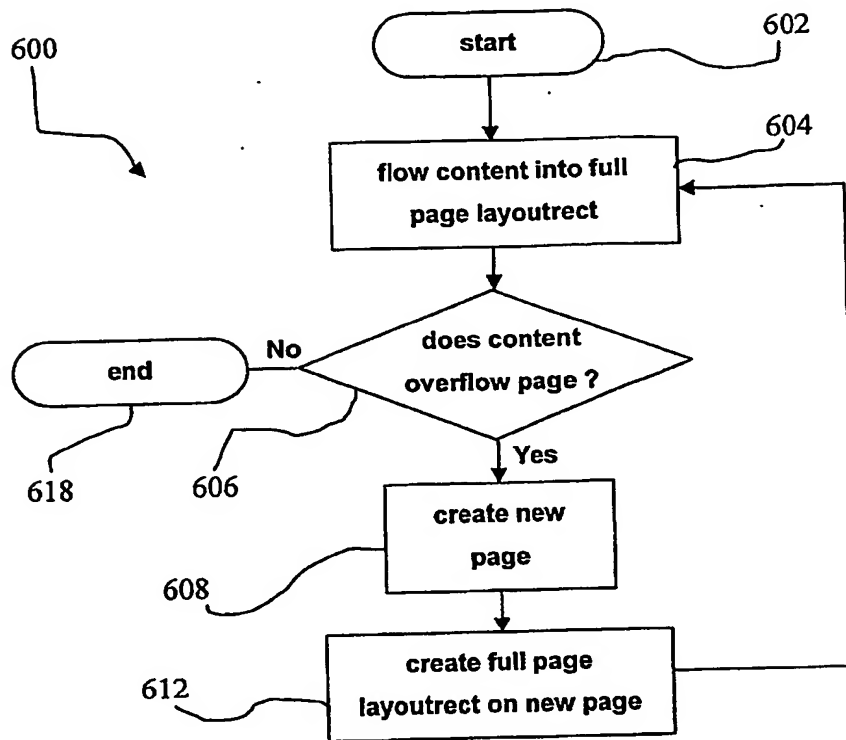


Fig. 4 (Prior Art)



**Fig. 5
(Prior Art)**

**Fig. 6 (Prior Art)**

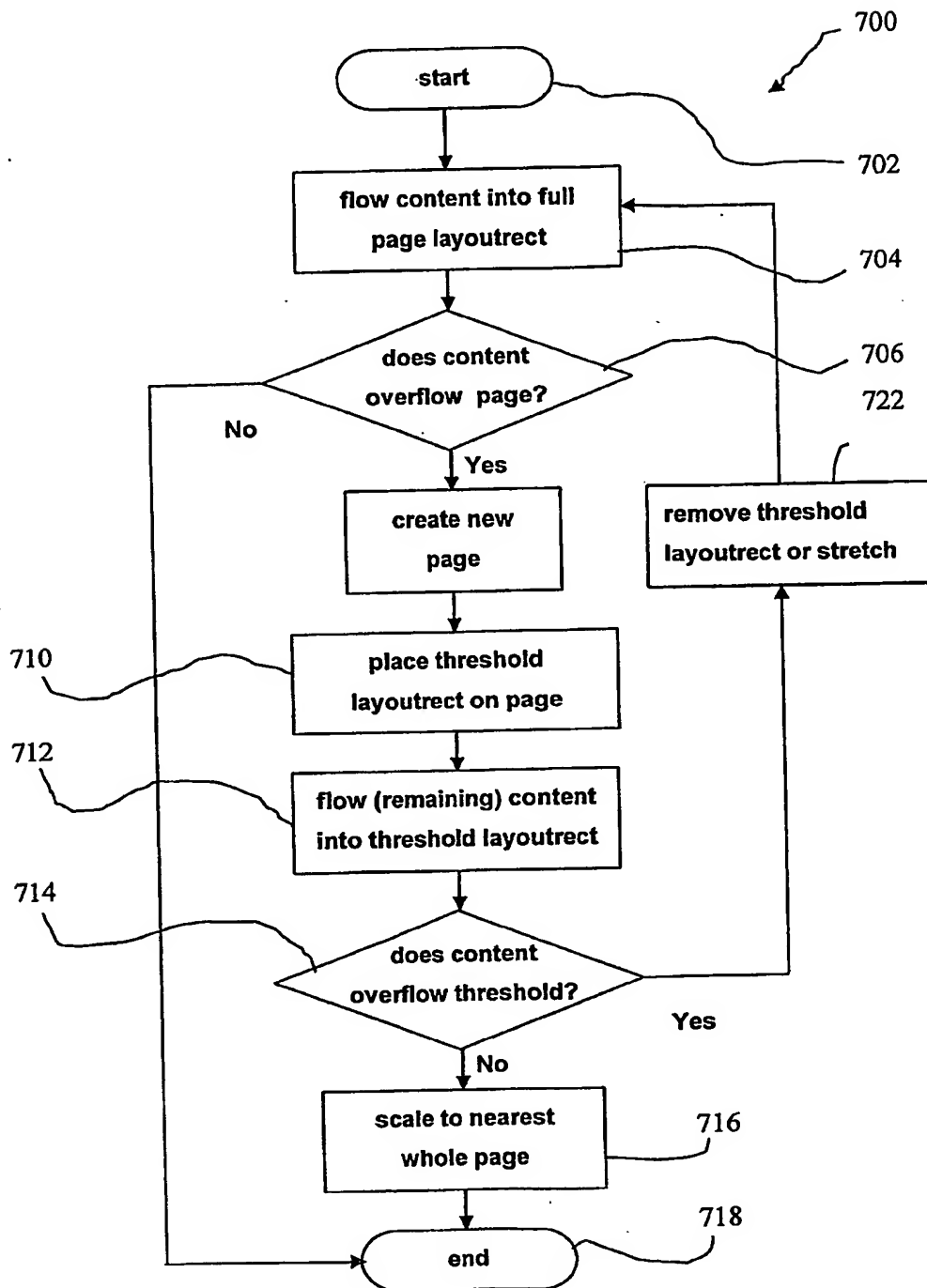


Fig. 7

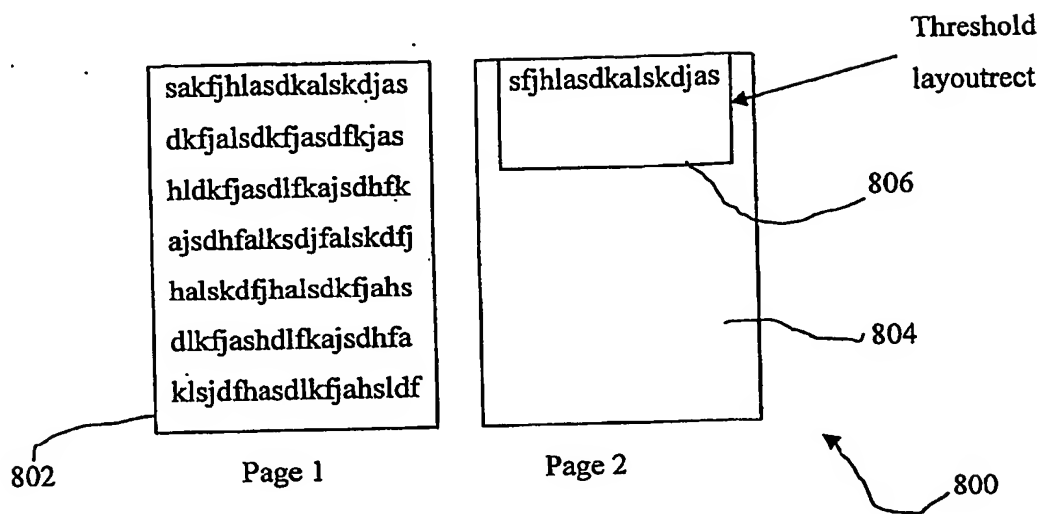


Fig. 8

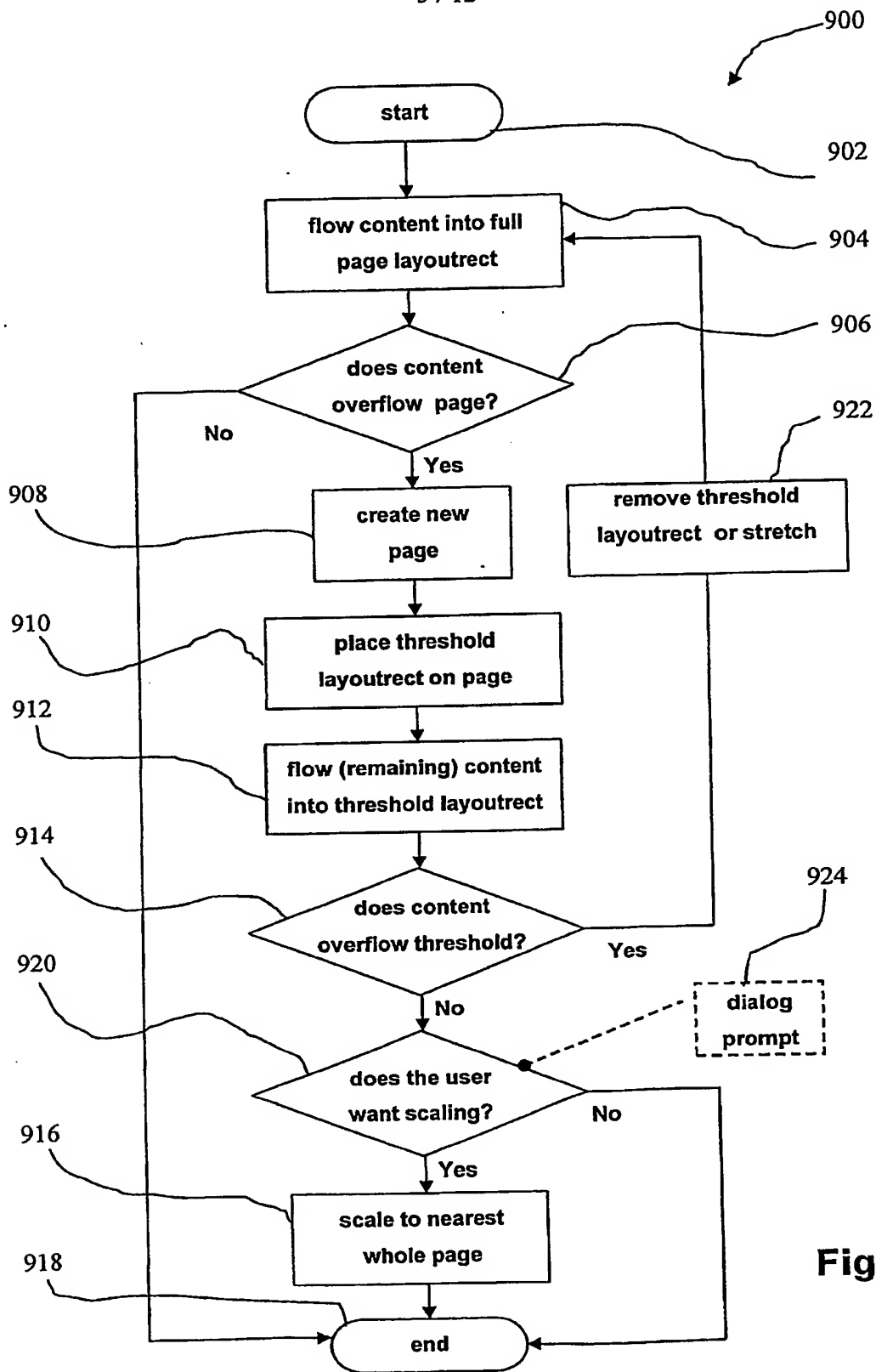
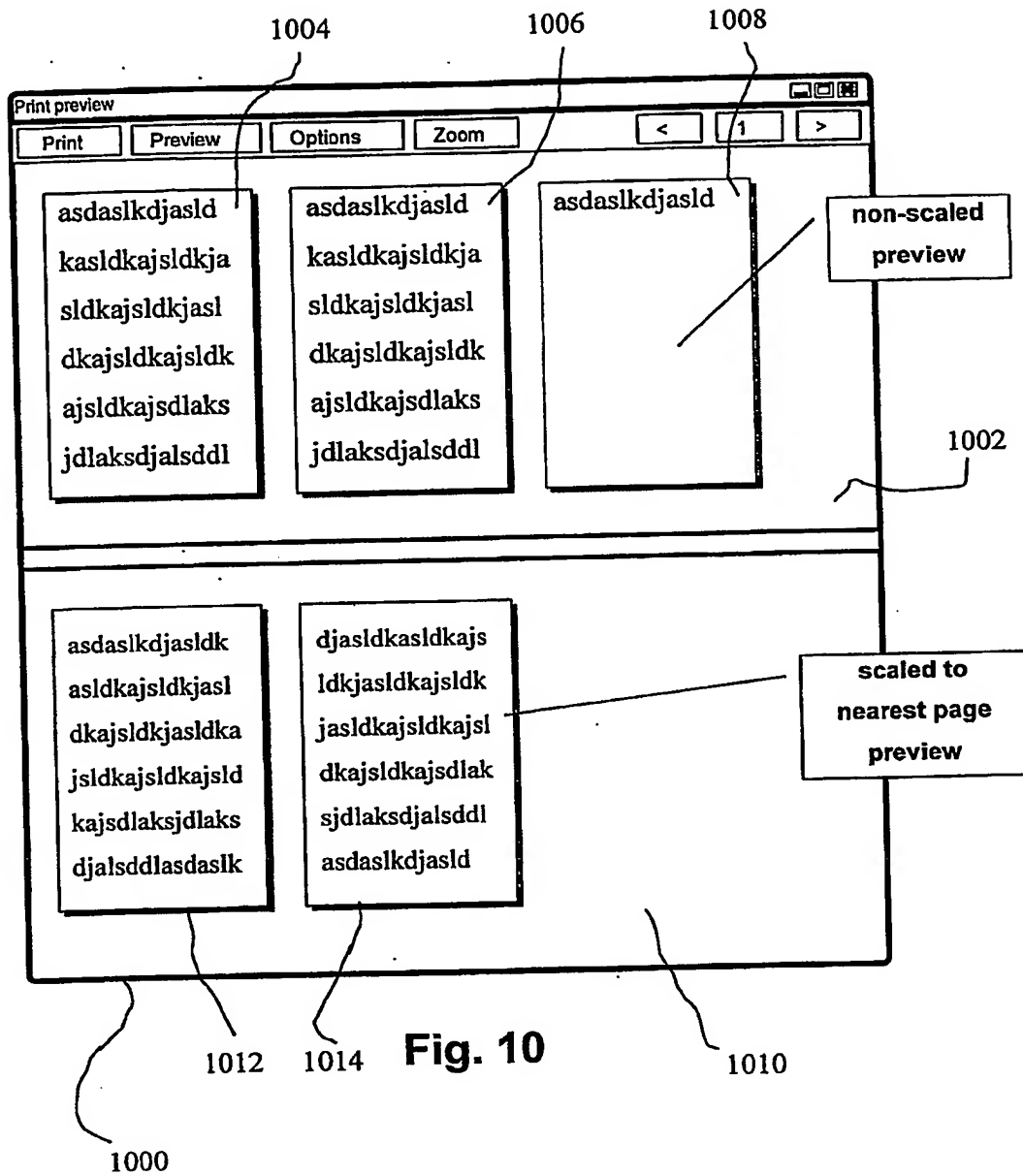


Fig. 9

**Fig. 10**

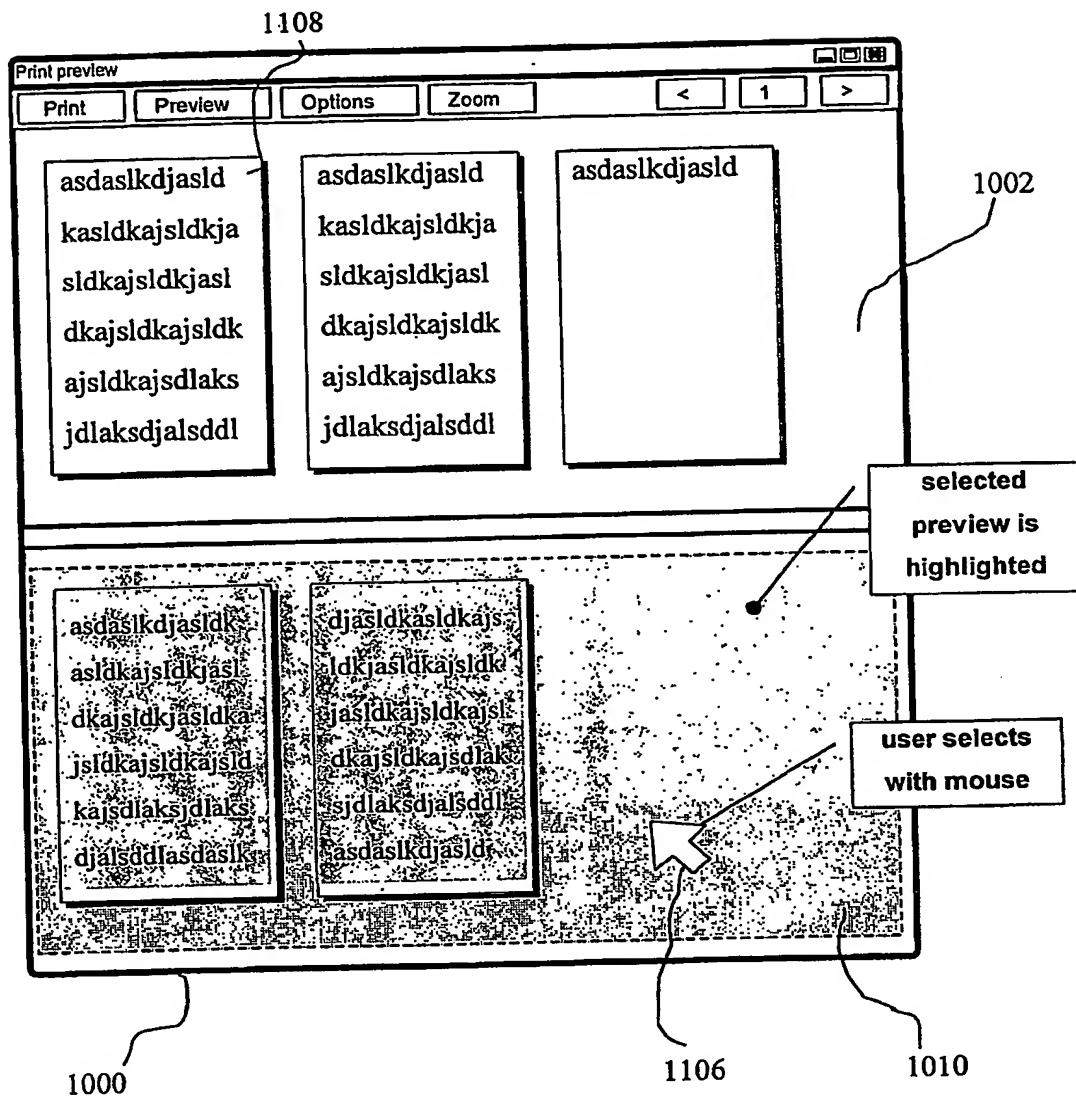


Fig. 11

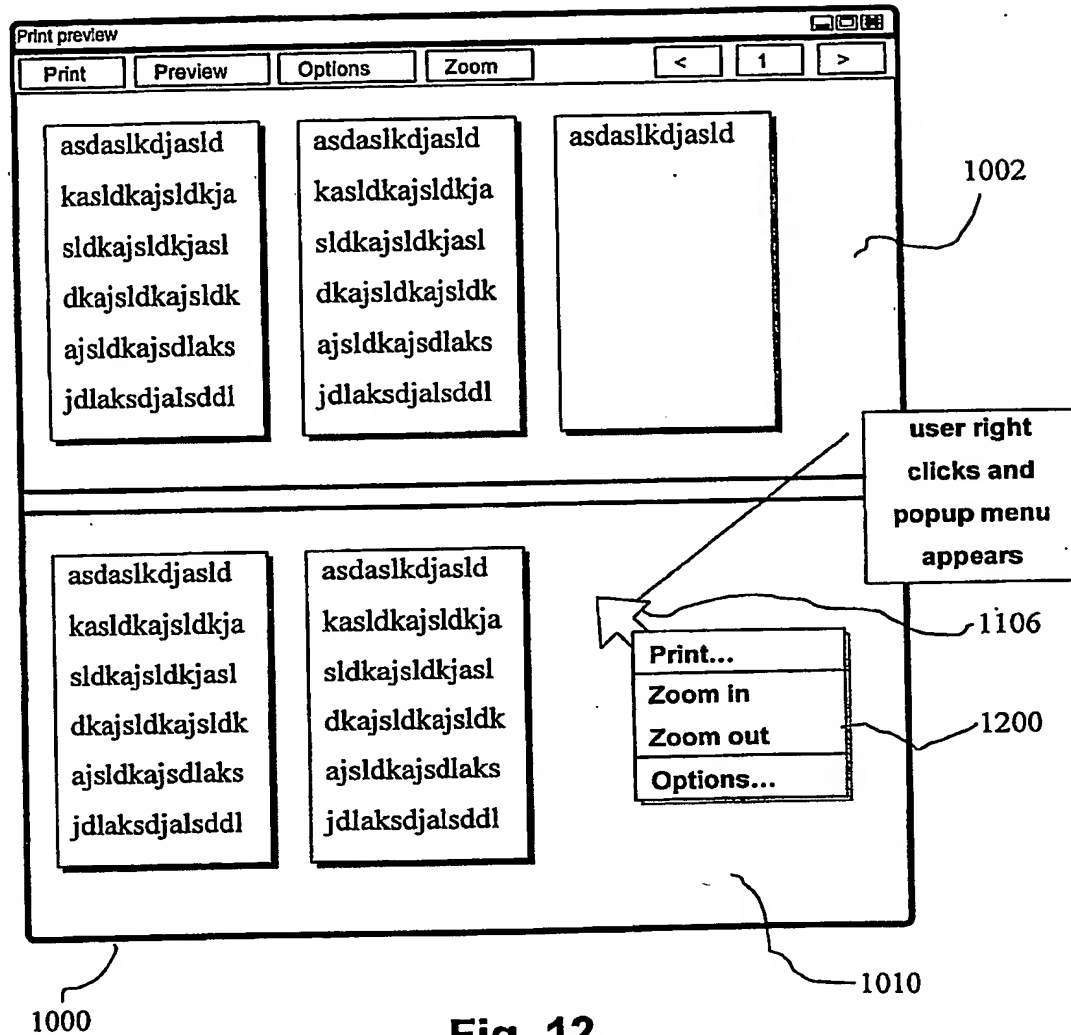


Fig. 12

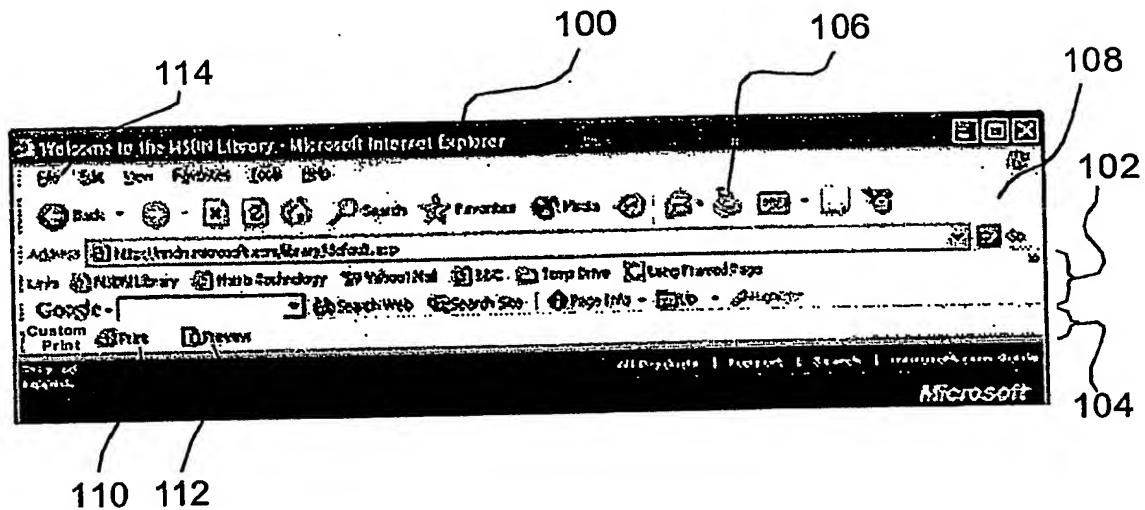


Fig. 1

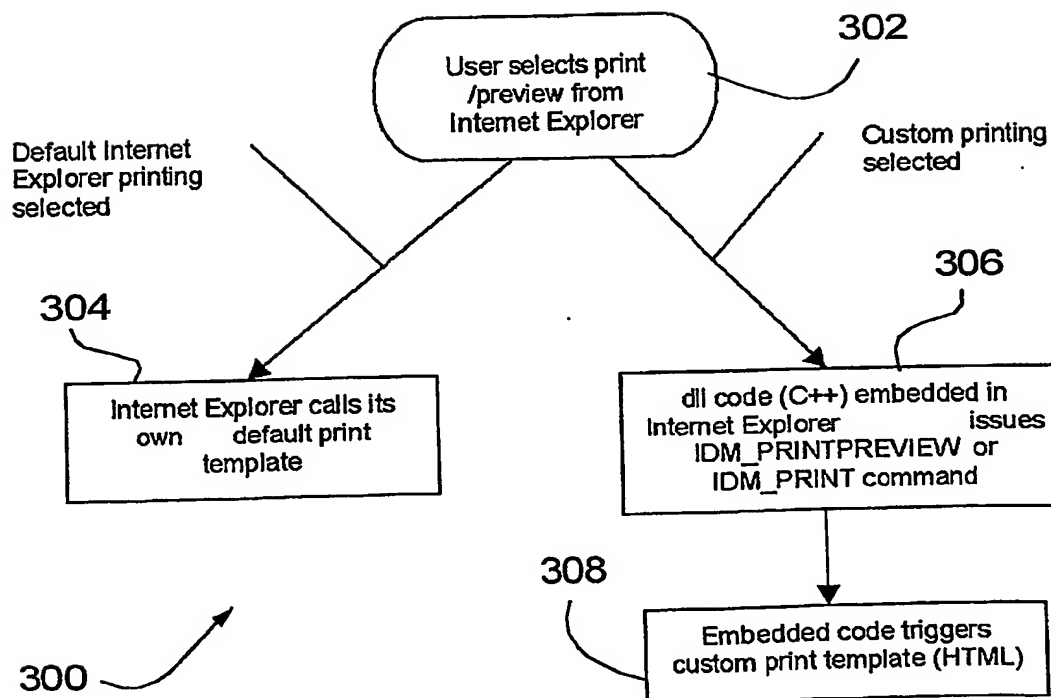
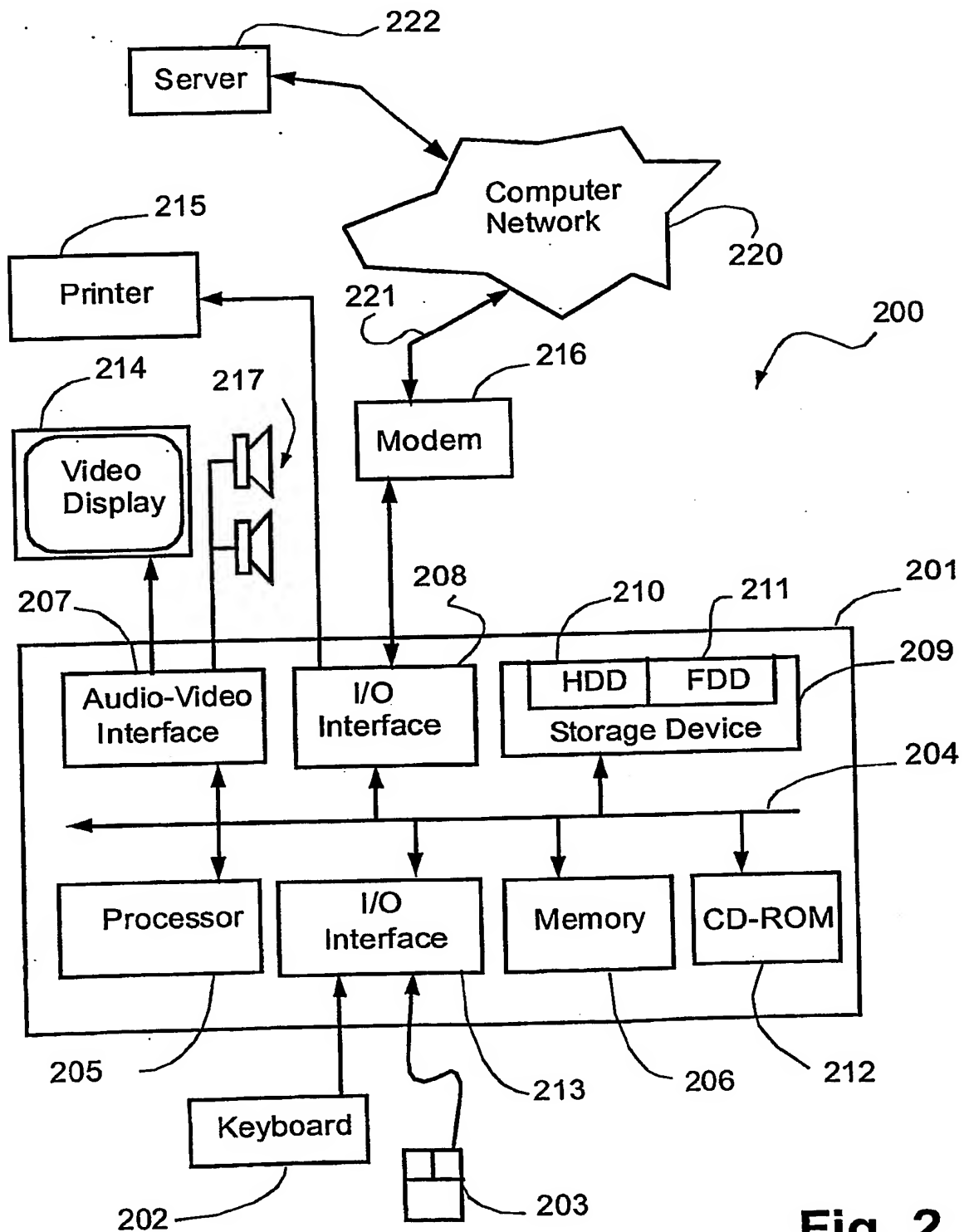


Fig. 3

**Fig. 2**

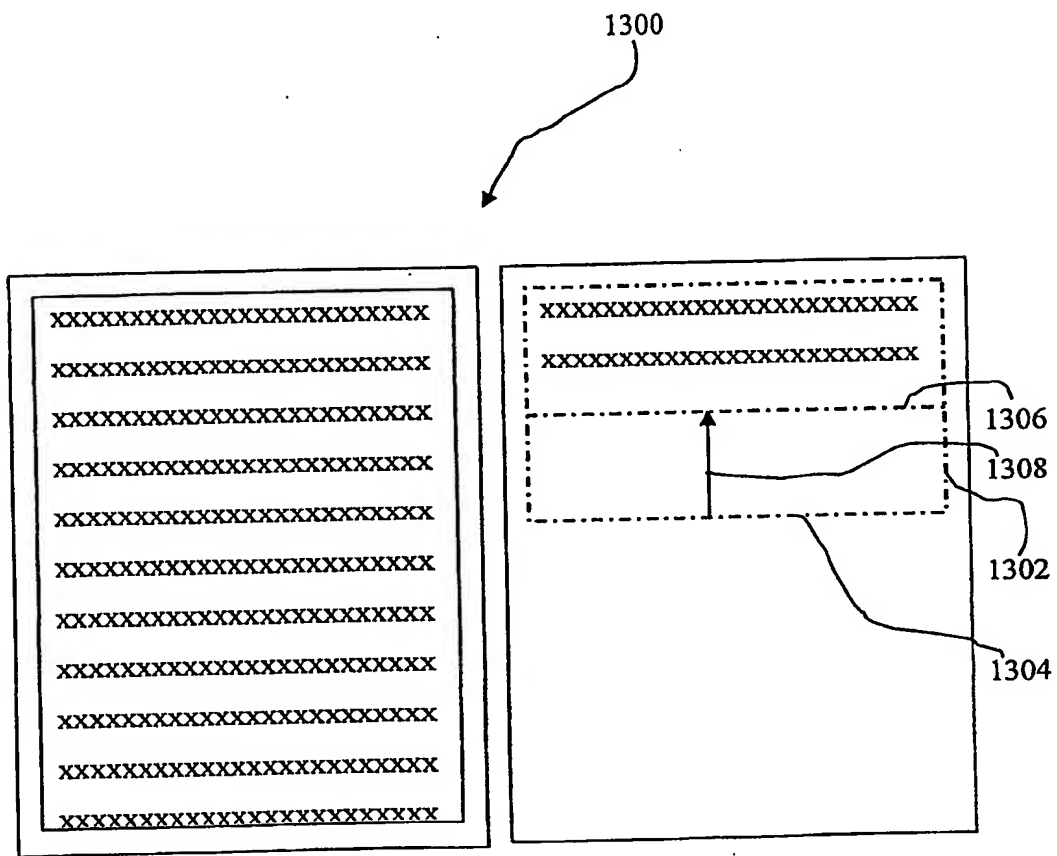


Fig. 13

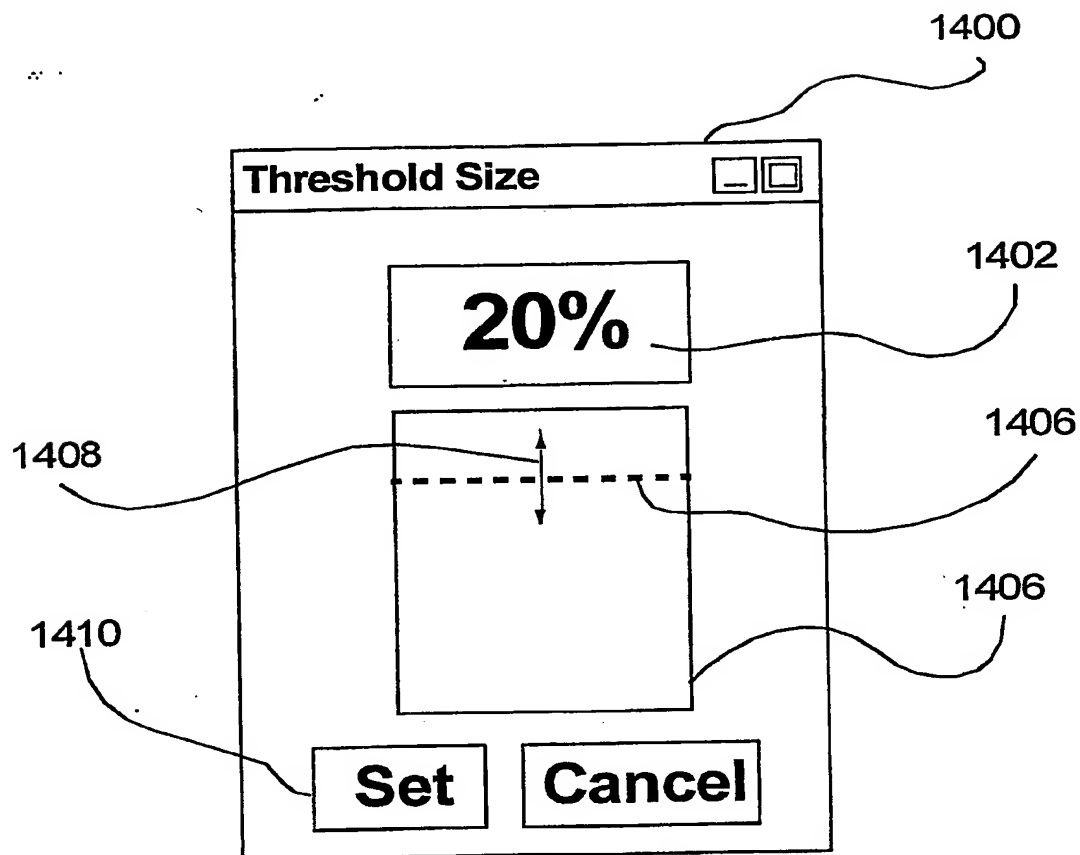


Fig. 14

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/AU04/001744

International filing date: 10 December 2004 (10.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: AU
Number: 2003906918
Filing date: 12 December 2003 (12.12.2003)

Date of receipt at the International Bureau: 04 January 2005 (04.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record.**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.